



HAL
open science

Parareal in time 3D numerical solver for the LWR Benchmark neutron diffusion transient model.

Mohamed-Kamel Riahi, Anne-Marie A.-M. Baudron, Jean-Jacques Lautard,
Yvon Maday, Julien Salomon

► **To cite this version:**

Mohamed-Kamel Riahi, Anne-Marie A.-M. Baudron, Jean-Jacques Lautard, Yvon Maday, Julien Salomon. Parareal in time 3D numerical solver for the LWR Benchmark neutron diffusion transient model.. 2011. hal-00722802v1

HAL Id: hal-00722802

<https://hal.science/hal-00722802v1>

Preprint submitted on 4 Nov 2012 (v1), last revised 13 Jan 2015 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Time-parareal parallel in time integrator solver for time-dependent neutron diffusion equation

J.-J. Lautard, A.-M. Baudron

CEA-DRN/DMT/SERMA, CEN-Saclay, 91191 Gif sur Yvette Cedex, France

Y. Maday¹,

*UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, and
Division of Applied Mathematics, Brown University, Providence, RI, USA.*

M.-K. Riahi¹,

UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France.

J. Salomon¹,

CEREMADE, Univ Paris-Dauphine, Pl. du Mal. de Lattre de Tassigny, F-75016, Paris France.

1

Abstract

The accurate knowledge of the time-dependent spatial flux distribution in nuclear reactor is required for nuclear safety and design. The motivation behind the development of parallel methods for solving the energy-, space-, and time-dependent kinetic equations is not only the challenge of developing a method for solving a large set of coupled partial differential equations, but also a real need to predict the performance and assess the safety of large commercial reactors, both these presently operating and those being designed for the future. Numerical results show the efficiency of the parareal method on large light water reactor transient model.

Keywords: Parareal in time algorithm, time-dependent neutron diffusion equations, reduced model, parallel computation, high performance computing.

2010 MSC: code, primary 65Y05; Secondary 35Q20.

☆

Email addresses: jean-jacques.lautard@cea.fr (J.-J. Lautard, A.-M. Baudron),
anne-marie.baudron@cea.fr (J.-J. Lautard, A.-M. Baudron), maday@ann.jussieu.fr (Y. Maday),
riahi@ann.jussieu.fr -/- riahi.mk@gmail.com (M.-K. Riahi), salomon@ceremade.dauphine.fr (J. Salomon)

URL: <http://www.ljll.math.upmc.fr/maday> (Y. Maday), <http://www.ljll.math.upmc.fr/riahi>
(M.-K. Riahi), <http://www.ceremade.dauphine.fr/~salomon> (J. Salomon)

1. Introduction

In this paper, we present a time-parallel algorithm that simulate the kinetic of neutron in a nuclear reactor. We consider a large set of coupled time-dependent partial differential equations (PDEs) that predict the nuclear engine behavior and in particular its energy production.

The flux distribution on the nuclear reactor is given by a time-dependent neutron transport equation. For reason of computing time, the time-dependent neutron diffusion equation is usually used to study the transient behavior of the flux. These diffusion equations are coupled with the dynamic of some *delayed neutrons*, which are called precursors.

The methods for numerically advancing the space-time diffusion group diffusion equations, along with their time dependent delayed neutron precursors counterparts, through time have been developed. For each of these considered methods, the system to be solved has been reduced to the matrix form. The H-method is an accurate and efficient finite difference scheme that has been employed in the numerical integration of the time-dependent group diffusion equations since the mid-1960s. First-order finite difference schemes such as the implicit (backward) Euler, explicit (forward) Euler, and Crank-Nicholson (central difference) schemes can be derived easily from the H-method in its general form. The time approximation used for the solution of the kinetic diffusion equation with the classical method or the improved quasi-static method [3] is based on an integral θ -scheme or a difference θ -scheme. The kinetics diffusion equations are solved with a time discretization using an integral θ -scheme both on the flux and the precursors equations (exact integration of the precursor equations with a linear expansion of the cross sections and polynomial representation of the flux). In this paper, for a sack of simplicity, we will consider difference θ -scheme for both neutron flux and it's time-dependent delayed precursor groups equation. Since not yet parallelization in time was not purely investigated with large scale Kinetic equations, our main topic in this paper is the time-parallel solving of the coupled equations present in the neutron model.

The knowledge of the energy of the nuclear reactor at forward long time is very important for nuclear safety, hence very long time machine simulation. In addition for a lack of availability of read-write memory in the sequential computers nowadays, it is relevant to often propose parallel methods, which solve these large scaled system, with massively parallel computers. Many successful works, has been done around parallelization of the approximated resolution of partial differential equation arising from neutron models. For instance [2] studies the static case i.e. Eigenvalues problems with space domain decomposition methods, and a very nice strategy [3], [4] uses quasi-stationary approach to accelerate the simulation. This approach opens some directions to parallelize the implementation.

In this paper we investigate the application of the parareal in time algorithm [5, 6] on neutron diffusion equation that governs the time-dependent flux distribution in the nuclear reactor. The parareal in time algorithm is an iterative scheme, that enables to improve computational time with parallel simulation. In several cases, parareal in time algorithm gives an impressive rate of convergence for the linear diffusion equations or more unexpected efficiency with non-linearity [7]. This algorithm is studied and shows stability and convergence results [8, 9, 10, 11, 12] particularly for diffusion system and others. Also it presents efficiency in parallel computer simulation. We find a variety [13, 14, 15, 9, 16, 17, 18, 19] of versions of this scheme that adapt [20, 21, 22, 23, 24, 25]

47 the original algorithm to tackle new settings. Furthermore the parareal algorithm can be
 48 easily coupled with other iterative schemes such as domain decomposition methods e.g.
 49 basic Schwartz algorithm or more complex one [26], and optimal control based descent
 50 algorithm [27, 28, 29].

51 The work in this paper is presented as follows: After this introduction, we present at
 52 the first section the partial differential equation that model the kinetic of neutron inside
 53 the nuclear reactor. We discuss that model, explain the real-physics and present the
 54 maths beside. The second section is devoted to numerical tools that we use to space-
 55 time discretize the problem. We introduce in the end of this section the parareal in time
 56 algorithm, which we adapt to the discretized problem. The parareal in time algorithm
 57 is therefore presented in a discrete way related to unification of variables that we solve
 58 all-at-once. We develop at the fourth section some approach that enables us to reduce
 59 the first model while remaining the reduced model close to the non-reduced one. That
 60 reduction is implemented in the coarse solver of the parareal in time algorithm that helps
 61 us to predict the solution for a forward time. We finally present and discuss the numerical
 62 experiments that show the speedup given when using the parareal in time algorithm.

63 2. Model

64 The neutron dynamics of a nuclear system (in absence of any external source) can
 65 be modeled by a time-dependent neutron diffusion equation (e.g. the equation (1))
 66 associated with the time-dependent delayed neutron precursor equations (e.g. the equa-
 67 tion (2)).

68 2.1. Time-dependent partial differential equation

69 The directional neutron flux solution of kinetic equation is a function at \vec{r} about point
 70 of reactor domain, in the interval “ dE ” about energy E , moving in the cone of directions $d\vec{\omega}$
 71 about direction of $\vec{\omega}$ at time t . For computational reasons, we use different simplifications.
 72 Concerning the energy variable, the time-dependent equations are usually solved using its
 73 multi-group formulation which is the basis for the vast majority of computer programs.
 74 By integrating in limited energy intervals from E_g to E_{g+1} obtained by subdividing the
 75 energy E from zero to infinity into \hat{g} intervals, a set of \hat{g} coupled differential equations.
 76 The second simplification concerns the direction variable. We look for the scalar flux
 77 which is an average flux on different directions. The scalar flux is solution of a simplified
 78 transport equation: the following diffusion equation

$$\left\{ \begin{array}{l} \frac{1}{V_g} \frac{\partial \phi_g(\vec{r}, t)}{\partial t} = \operatorname{div}(D_g \vec{\nabla} \phi_g(\vec{r}, t)) - \Sigma_t^g \phi_g(\vec{r}, t) + \chi_g^p \sum_{g'} (1 - \beta_{g'}) \nu \Sigma_f^{g'} \phi_{g'}(\vec{r}, t) \\ \quad + \sum_{g'} \Sigma_s^{g'} \phi_{g'}(\vec{r}, t) + \sum_{k=1}^K \chi_{k,g}^d C_k(\vec{r}, t), \quad \vec{r} \in \Omega \subset \mathbb{R}^3. \\ \phi_g(\vec{r}, t) = 0 \quad \text{on the boundary of the nuclear reactor} \\ \phi_g(\vec{r}, 0) = \phi_g^0(\vec{r}) \quad \text{a given initial value} \end{array} \right. \quad (1)$$

79 The delayed neutron precursor satisfy :

$$\frac{\partial C_k}{\partial t}(\vec{r}, t) = -\lambda_k C_k(\vec{r}, t) + \sum_{g'}^{\hat{g}} \beta_{k,g'} \nu \Sigma_f^{g'} \phi_{g'}(\vec{r}, t) \quad (2)$$

80 In Equations (1) and (2) one reads:

- 81 V_g is the neutron velocity in energy group g ,
- 82 ϕ_g is the neutron scalar flux in energy group g ,
- 83 D_g is the diffusion coefficient in energy group g ,
- 84 $\Sigma_t^g, \nu\Sigma_f^g$ are the total and production cross-section in energy group g ,
- 85 $\Sigma_s^{g'g}$ is the transfer cross-section from energy group g' to g ,
- 86 χ_g^p, χ_{kg}^d is the fission spectra of prompt and delayed neutrons, respectively C_k is the
- 87 concentration of precursor group k ,
- 88 β_{kg}, λ_k are the delay fraction in energy group g and decay constant of precursor group k ,
- 89 β_g is the total delay fraction in energy group g ($\beta_g = \sum_k \beta_{kg}$).

90 The equation (1) is derived from the transport equation by expanding into spherical
 91 harmonics and integrated over all directions $\vec{\omega}$, neglecting several terms. For each energy
 92 group $g = 1, \dots, \hat{g}$ and $\vec{r} \in \Omega \subset \mathbb{R}^3$, the equations are a set of the three-dimensional,
 93 multi-energy-group neutron kinetics equations, which are derived from (1).

94 2.2. Boundary condition in symmetrical physical core

In what follows, for reason of simplicity, we restrict our simulation for a quarter of the domain (nuclear reactor). Consequently, we consider Neuman boundary condition at the symmetry axes. That translate some null outcome across theses interfaces. The boundary condition is thus:

$$\frac{\partial \phi}{\partial \vec{n}}(\vec{r}, t) = 0 \quad \text{on the interior interfaces.}$$

95 Such a way, the rest of the domain (nuclear reactor) is simply deduced by symmetry.

96 2.3. Initial condition determination

For the initial condition, we consider the stationary version of the equation (7). We use the concept of *reactivity* which is related to the energetic behavior of the nuclear reactor. We focus on the distribution of the flux when the nuclear reactor is stable. This distribution is thus said *stable state*. In order to calculate this stable state, we introduce an eigenvalues problem on *effective reactivity* denoted by a k_{eff} factor. This factor indicates the greatest generalized eigenvalue of the two matrices $\hat{F} := \Sigma_f^f$ and \hat{A} (see Eq (9) below) of *production* and *absorption* respectively. The production matrix characterizes the production of the neutrons by fission, whereas absorption matrix characterize the escapes/leaks. The eigenvalues problem is as follows:

$$\hat{A}\Phi = \frac{1}{k_{eff}}\hat{F}\Phi.$$

97 We use finally an the power algorithm to compute the k_{eff} factor.

98 The nuclear core may have three states of energy (*subcritical, critical and supercritical*).
 99 These three states are related to the density of the neutrons diffusing in the core. That
 100 density is due to neutron production by the chain reaction whereas the disappearance of
 101 the neutrons by absorption and/or some leaks. The core is said *critical* if the neutron

102 outcome is null. It means: at every moment the number of produced neutrons is equal to
 103 the number of disappeared ones. The neutron outcome inside core at equilibrium obeys:

$$\text{production}_{fission} + \text{source}_{external} = \text{absorption} + \text{leaks}. \quad (3)$$

104 Neutrons injected into the system constitute the source of the reaction, absorption cans
 105 be important, the escape/leaks represents the neutrons leaving the system.

106 The multiplicative factor in a medium without escape/leaks, noted by k_∞ is defined by:

$$k_\infty := \frac{\text{neutron account at time } t}{\text{neutron account at time } t-1}. \quad (4)$$

107 In a bounded domain, The effective multiplicity factor k_{eff} is as follows:

$$k_{eff} = \frac{k_\infty}{1 + \text{leaks}}. \quad (5)$$

This coefficient should be very close to unity in order to guarantee a state of equilibrium. Thus, if $k_{eff} < 1$ the state of the core is said subcritical, else if $k_{eff} > 1$ the state is supercritical and if $k_{eff} = 1$ the state is critical.

In order to determine such a state, it is essential to evaluate the stationary state without external source i.e. the unique source considered is fission.

The stationary Boltzmann equation (6) admits solutions only in the critical case. In this particular framework, we consider the fission source term extracted from Equation (1):

$$\chi^{(g)} \sum_{m=1}^{\hat{g}} \nu^{(g')} \sigma_f^{(g')} \phi^{(g')}(\vec{r}, t),$$

and absorption term (diffusion and scattering) also from Equation (1):

$$-\text{div}(D(\vec{r}, t) \nabla \phi^{(g)})(\vec{r}, t) + \sigma_t^{(g)}(x) \phi^{(g)}(\vec{r}, t) - \sum_{g'=1}^{\hat{g}} \sigma_s^{(g \rightarrow g')} \phi^{(g')}(\vec{r}, t).$$

108 The input data of modeling do not necessary characterizing a critical state and if nec-
 109 essary, we force our system becoming critical by adding the k_{eff} factor in the stationary
 110 version of Equation (1). This equation becomes then:

$$\begin{aligned} & -\text{div}(D(\vec{r}, t) \nabla \phi^{(g)})(\vec{r}, t) + \sigma_t^{(g)}(x) \phi^{(g)}(\vec{r}, t) - \sum_{g'=1}^{\hat{g}} \sigma_s^{(g \rightarrow g')} \phi^{(g')}(\vec{r}, t) \\ & = \\ & \frac{1}{k_{eff}} \chi^{(g)} \sum_{g'=1}^{\hat{g}} \nu^{(g')} \sigma_f^{(g')} \phi^{(g')}(\vec{r}, t). \end{aligned} \quad (6)$$

111 One thus bring himself back to a generalized eigenvalues problem that consists to look
 112 for k_{eff} factor for which the critical outcome equation (3) holds.

113 We thus use the power method to compute the largest eigenvalues. The corresponding
 114 eigenvector will be the neutron flux distribution that is the solution of the stationary
 115 problem (6).

116 **3. Numerical methods and time-parallel-solver**

117 For the convenience of presentation, we gather variables of various groups of energy
 118 both on neutron flux and delayed neutrons all at ones sought a vectorial form within
 119 the continuous framework. Then, we proceed to the time-space discretization, in order
 120 to have finally a block matrix structure representing the linear system that solves the
 121 problem at each time step.

122 This procedure of unification facilitates the application of the parareal in time algorithm
 123 in the resolution of the neutron model. Indeed, in a practical point of view the vectorial
 124 variable form makes building of the time-propagator easier.

125 *3.1. Space discretization*

We assume that the neutronic flux i.e. ϕ modeled by Equations (1)-(2) belongs to $C^0([0, T]; H_0^1(\Omega))$ and refer to [30, 31, 32, 33] for more theoretical results. We denote by $H_0^1(\Omega)$ the Hilbert space defined as follows:

$$H_0^1(\Omega) := \{u \in L^2(\Omega), \text{ such that } \nabla u \in L^2(\Omega), u = 0 \text{ on } \partial\Omega\},$$

126 and denote by $\langle \cdot, \cdot \rangle$ its scalar product.

In order to discretize the problem, we use finite elements space approximation associated with the weak formulation of the flux and the concentration of precursors. We note that the flux belongs to $H := H_0^1(\Omega)$ and the precursors belong to $U := L^2(\Omega)$ (see [35]), so by meshing the domain $\Omega \subset \mathbb{R}^3$ with an uniform triangulation $\mathcal{T}_h(\Omega)$, one can define finite dimensional Hilbert space $H_h \subset H$ (respectively $U_h \subset U$), with associated standard basis functions $(\psi_i)_{i=1}^q$ (respectively $(\phi_i)_{i=1}^p$). In addition one has:

$$\begin{aligned} H_h &:= \{u_h \in C^0(\Omega), u_{h\mathbf{K}} \in \mathbb{P}_1, \forall \mathbf{K} \in \mathcal{T}_h\}, \\ U_h &:= \{u_h \in C^0(\Omega), u_{h\mathbf{K}} \in \mathbb{P}_0, \forall \mathbf{K} \in \mathcal{T}_h\}. \end{aligned}$$

The physical problem admits a dynamic geometry because energy control rods are moving as well as the reaction proceeds. As the cross-sections are constant in each domain, it is more judicious to attach their dependence in time to the movement of those rods. These coefficients belong to

$$L^\infty(\Omega) := \{u : \Omega \rightarrow \mathbb{R}, \exists c > 0, |u(\vec{r})| \leq c, \forall \vec{r} \in \Omega\},$$

127 of which we consider an approximation \mathbb{P}_0 for the discrete forms.

128

129 For the deterministic model of the kinetics equation, the discretized cross sections are supposed to be constant on each elements of the mesh (called media).

130 In order to indicate the space-dependency on the cross-sections in their discretized notations, we use the notation with the hat symbol (i.e.: “ $\hat{\cdot}$ ”) over each vector of nodal coefficients, which is identified to its continuous notation.

131 By convenience, a block matrix belongs to $\mathbb{R}^{(ab) \times (cd)}$ means that it has a block rows and c block columns of matrices belonging to \mathbb{R}^{bd} i.e. b rows and d columns.

132 Let us consider $\hat{\Phi}$ and $\hat{\mathbf{c}}$ the nodal representation of first derivative in time of the neutron flux and the delayed precursors concentration respectively. The discrete multi-

137

138 group unified equation governing the neutron flux is as follows:

$$\mathcal{M}_h^v \dot{\Phi}(t) = \mathcal{A}_h^d \Phi(t) - \Sigma_h \Phi(t) + \mathcal{R}_h \mathbf{c}(t), \quad (7)$$

In Equation (7) $\mathcal{M}_h^v := \text{Blockdiag}(M_h^v) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}$, with $M_h^v \in \mathbb{R}^{\hat{q}\hat{q}}$, and $\mathcal{A}_h^d := \text{Blockdiag}(A_h^d) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}$. The matrix $\Sigma_h \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}$, the matrix $\mathcal{R}_h := \text{Blockdiag}(R_h) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}$. Where $(M_h^v)_{i,j} := VM_h$, on the other hand $(M_h)_{i,j} := \langle \psi_i, \psi_j \rangle$, $(A_h^d)_{i,j} := \langle \bar{D}_i \nabla \psi_i, \nabla \psi_j \rangle$, where \bar{D}_i is the nodal representation of D (constant by element as \mathbb{P}_0), $(R_h)_{i,j} := \langle (\widehat{\chi_d \mu})_i \psi_i, \psi_j \rangle$. The matrix Σ_h of the cross section are defined as :

$$\Sigma_h := \Sigma_h^t - \Sigma_h^s + (Id - [\widehat{\beta}]) \Sigma_h^f,$$

where,

$$\begin{aligned} \Sigma_h^t &:= \text{Blockdiag}(\Sigma_h^t) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}, \\ \Sigma_h^s &:= \text{Blockdiag}(\Sigma_h^s) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}, \\ \Sigma_h^f &:= \text{Blockdiag}(\Sigma_h^f) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}, \\ [\widehat{\beta}] &:= \text{Blockdiag}(\beta^{(i)} I_{\mathbb{R}^{\hat{g}\hat{g}}}) \in \mathbb{R}^{(\hat{g}\hat{q}) \times (\hat{g}\hat{q})}, \end{aligned}$$

with

$$\begin{aligned} (\Sigma_h^t)_{i,j} &:= \langle (\widehat{\sigma}_t)_i \psi_i, \psi_j \rangle \in \mathbb{R}^{\hat{q}\hat{q}}, \\ (\Sigma_h^s)_{i,j} &:= \langle (\widehat{\sigma}_s)_i \psi_i, \psi_j \rangle \in \mathbb{R}^{\hat{q}\hat{q}}, \\ (\Sigma_h^f)_{i,j} &:= \langle (\widehat{\chi_p v \sigma_f})_i \psi_i, \psi_j \rangle \in \mathbb{R}^{\hat{q}\hat{q}}, \end{aligned}$$

139 where the sections $\widehat{\sigma}_t$, $\widehat{\sigma}_s$ et $\widehat{\chi_p v \sigma_f}$ are the finite elements nodal representations with \mathbb{P}_0
 140 approximation, in addition $(\widehat{\sigma}_t)_i$, $(\widehat{\sigma}_s)_i$ and $(\widehat{\chi_p v \sigma_f})_i$ referred to the tetrahedral indexed
 141 by i .

142 The discrete equation that governs the concentration of precursors \mathbf{c} is:

$$\mathcal{M}_h \dot{\mathbf{c}}(t) = \mathcal{N}_h \mathbf{c}(t) + [\widetilde{\beta}] \Sigma_h^f \Phi(t), \quad (8)$$

143 where $\mathcal{M}_h^c \in \mathbb{R}^{(\hat{k}\hat{p}) \times (\hat{k}\hat{p})}$, $\mathcal{M}_h^c := \text{Diag}(M_h^c)$, with $(M_h^c)_{i,j} := \langle \phi_i, \phi_j \rangle$. The matrix $\mathcal{N}_h :=$
 144 $\text{Blockdiag}(N_h^k) \in \mathbb{R}^{(\hat{k}\hat{p}) \times (\hat{k}\hat{p})}$ with $(N_h^k)_{i,j} := (\mu_i^{(k)} \phi_i, \phi_j)$, and $[\widetilde{\beta}] \in \mathbb{R}^{(\hat{k}\hat{p}) \times (\hat{k}\hat{p})}$, such that $([\widetilde{\beta}])_{k,g} :=$
 145 $(\widetilde{\beta}^{(g)}) \in \mathbb{R}^{\hat{p}\hat{p}}$ où $(\widetilde{\beta}^{(g)})_{i,j} := \beta^{(g,k)} I_{\mathbb{R}^{\hat{p}\hat{p}}}$.

146 In the case where we do not have initial condition for Equation (7), we can build it
 147 using some techniques that we develop hereafter.

148 Let us now describe, with more details, implied matrices in the problem. The matrices
 149 disappearance (absorption+leaks) are described as follows:

$$\widehat{\mathbf{A}} := \mathcal{A}_h^d - \Sigma_h^t + \Sigma_h^s, \quad (9)$$

and the matrix related to the fission source term $\widehat{\mathbf{F}}$. Thanks to those matrices, we can

compute the generalized eigenvalue k_{eff} of the couple $(\widehat{\mathbf{A}}, \widehat{\mathbf{F}})$, which satisfies:

$$\widehat{\mathbf{A}}\Phi = \frac{1}{k_{eff}}\widehat{\mathbf{F}}\Phi,$$

150 where Φ is the eigen-flux associated with k_{eff} . The algorithm hereafter (power algorithm)
 151 gives us the eigenvalue of the largest modulus. The space norm $\|\cdot\|_2$ is associated to the
 152 scalar product $\langle \cdot, \cdot \rangle_{\mathbb{R}^{3g}}$.

Algorithm 1: Fission source problem using power Algorithm

Input: Initial flow vector Φ^0 , tolerance $\tilde{\epsilon}$;
 Initialize the source term $\mathcal{S}^0 = \widehat{\mathbf{F}}\Phi^0$; $l \leftarrow 0$;
repeat
 1. Solve iteratively : $\widehat{\mathbf{A}}\Phi^{l+1} = \mathcal{S}^l$;
 2. Normalize $\tilde{\Phi}^{l+1} := \Phi^{l+1}/\|\Phi^{l+1}\|_2$;
 3. Evaluate the relative error $Err := \|\tilde{\Phi}^{l+1} - \tilde{\Phi}^l\|_2/\|\tilde{\Phi}^{l+1}\|_2$;
 4. Update the source : $\mathcal{S}^{l+1} = \widehat{\mathbf{F}}\tilde{\Phi}^{l+1}$;
 $l \leftarrow l + 1$;
until $Err \leq \tilde{\epsilon}$;
Result: $\Phi := \tilde{\Phi}^l$ & $k_{eff} := \|\Phi^l\|_2$;

153

Using Algorithm 1, we can build the initial condition $\Phi^{(g, \star)}$ of the neutronic flux (including all energy groups). Then using the formula

$$c^{(k)}(\vec{r}, 0) := \frac{1}{\mu^{(k)}} \sum_{g'=1}^{\hat{g}} \beta^{(k, g')} \nu \sigma_f^{(g')} \Phi^{(g')},$$

154 Therefore, we are able to build the initial condition related to the concentrations of
 155 precursors groups.

156 *3.2. Time discretization*

We denote by Φ_{i+1} the approximation of $\Phi(t_{i+1})$ obtained with the time-discretization θ -scheme.

$$\begin{aligned} \mathcal{M}_h^v \Phi_{i+1} - \mathcal{M}_h^v \Phi_i &= \tau \theta (\mathcal{A}_h^d \Phi_{i+1} - \Sigma_h \Phi_{i+1} + \mathcal{R}_h \mathbf{c}_{i+1}) \\ &\quad + \tau (1 - \theta) (\mathcal{A}_h^d \Phi_i - \Sigma_h \Phi_i + \mathcal{R}_h \mathbf{c}_i), \end{aligned} \quad (10)$$

and,

$$\mathcal{M}_h^k \mathbf{c}_{i+1} - \mathcal{M}_h^k \mathbf{c}_i = \tau \theta (\mathcal{N}_h \mathbf{c}_{i+1} + [\widetilde{\beta}] \Sigma_h^f \Phi_{i+1}) + \tau (1 - \theta) (\mathcal{N}_h \mathbf{c}_i + [\widetilde{\beta}] \Sigma_h^f \Phi_i), \quad (11)$$

157 where $\theta \in [0, 1]$ and τ is the time step.

158 We denote by $X_i := [\Phi_i^T, \mathbf{c}_i^T]^T$ the vector that represents the solution at time t_i . The

159 previous iterative formulas (10)-(11) can be written in the following matrix form:

$$\begin{bmatrix} \Phi_{i+1} \\ \mathbf{c}_{i+1} \end{bmatrix} = \begin{bmatrix} D_{II} & D_{IJ} \\ D_{JI} & D_{JJ} \end{bmatrix}^{-1} \begin{bmatrix} B_{II} & B_{IJ} \\ B_{JI} & B_{JJ} \end{bmatrix} \begin{bmatrix} \Phi_i \\ \mathbf{c}_i \end{bmatrix}, \quad (12)$$

where

$$\begin{aligned} D_{II} &:= \mathcal{M}_h^v + \tau\theta(\Sigma_h - \mathcal{A}_h^d), & B_{II} &:= \mathcal{M}_h^v + \tau(1 - \theta)(\mathcal{A}_h^d - \Sigma_h), \\ D_{IJ} &:= -\tau\theta\mathcal{R}_h, & B_{IJ} &:= \tau(1 - \theta)\mathcal{R}_h, \\ D_{JI} &:= -\tau\theta[\widetilde{\beta}]\Sigma_h^f, & B_{JI} &:= \tau(1 - \theta)[\widetilde{\beta}]\Sigma_h^f, \\ D_{JJ} &:= \mathcal{M}_h^k - \tau\theta\mathcal{N}_h, & B_{JJ} &:= \mathcal{M}_h^k + \tau(1 - \theta)\mathcal{N}_h, \end{aligned}$$

160 which we simply denote by:

$$X_{i+1} = EX_i, \quad (13)$$

where the block matrix E is :

$$E := \begin{bmatrix} D_{II} & D_{IJ} \\ D_{JI} & D_{JJ} \end{bmatrix}^{-1} \begin{bmatrix} B_{II} & B_{IJ} \\ B_{JI} & B_{JJ} \end{bmatrix}.$$

161 In what follows, we describe the construction of the sequential 3D solver. Since control
162 energy rods are moving as physical time progress, the space domain properties also vary
163 with respect to the positions of rods and fuels in the nuclear reactor. Hence the matrix
164 E is time-dependent, sequential and requires an update/assembly at each time step. The
165 construction of the serial solver can so be achieved with following three steps:

166 Step 1 : localize the absorber rods according to the chronology of the corresponding rod
167 group position, then calculate the cross sections.

168 Step 2 : construct and assemble the block matrices.

169 Step 3 : solve the system e.g. with the GMRES algorithm [34].

170 Let us describe now in more details some characteristics of our solver and the implementa-
171 tion we define in the context of an international standard Benchmarck [35] . In particular
172 we will detail adaptation of cross-sections to the mesh generation and managing dynam-
173 ical geometry. Let us now discuss two procedures: the first one uses an adaptation of the
174 number of the layers (on meshes), on the “z” direction, according to chosen small times
175 steps. The second one, what we investigate in this paper is based on the interpolation
176 of the Heaviside functions according to the celerity of control rods. We use tetrahedron
177 elements generated by the scientific computation software FreeFem++ [1] and we carry
178 out the rods movements by interpolating the domain indicators, which characterizes the
179 parts of the finite elements (tetrahedron) occupied by rods and/or fuel. These indica-
180 tors are used to distinguish constant physical coefficients in each medium/domaine (see
181 tables (3)–(4)) that are approximated by \mathbb{P}_0 finite elements.

182 In the setting of our solver, the construction of block matrices requires a construction
183 of Heaviside functions, which identify the mediums, by the intuitive rule:

$$\text{Presence of rods} \implies \text{Absence of fuel.} \quad (14)$$

The 3D-mesh (Figure on the side) represents $\frac{1}{4}$ nuclear core (front view). The rest of the core is deduced by symmetry.

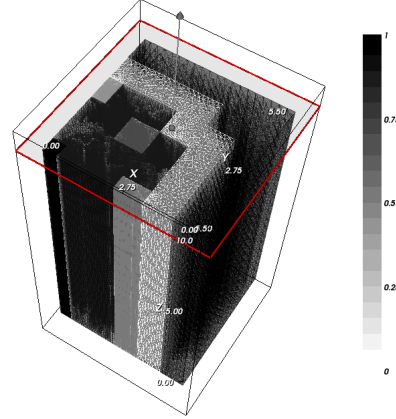


Figure 1: 3D mesh representation of $\frac{1}{4}$ of the nuclear core .

184 In the framework of \mathbb{P}_0 finite elements approximation on each tetrahedron, the Heaviside
 185 functions of the mediums worth “1” or “0”. However, this modeling is not sophisticated
 186 enough because it produces some undesirable oscillation.

187 Let us suppose that at time t_1 the rods start to be at the same level as a tetrahedron
 188 and than to times t_2 passes to the following tetrahedron. For the period of time $t_2 - t_1$
 189 the rods are located in specific tetrahedrons (see Figure 3).

190 Our approach consists in, initially measuring the rate of sinking of that rods, then
 191 assigning a value between “0” and “1” to the Heaviside functions. This method takes into
 192 account the vertical movement of the rods. Once the Heaviside functions of the rods are
 assembled, those of fuel are deduced according to the rule (14). We are now considered

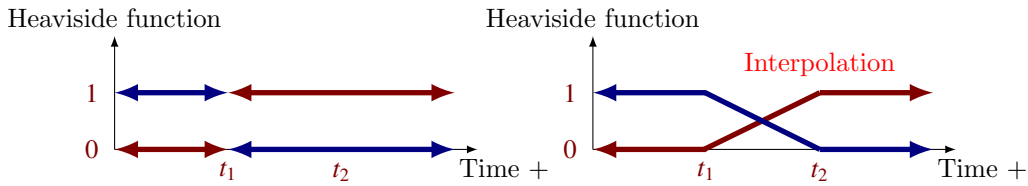


Figure 2: Heaviside function indicates the rate of sinking of the rods (red) and of the Fuel (blue) on a tetrahedron.

193 with a an adaptive linear system such that:
 194

$$X_{i+1} = E_i X_i, \quad (15)$$

195 which motivate parallel computation to increase serial simulation cost in ordinary ma-
 196 chine, and thus improve computational time.

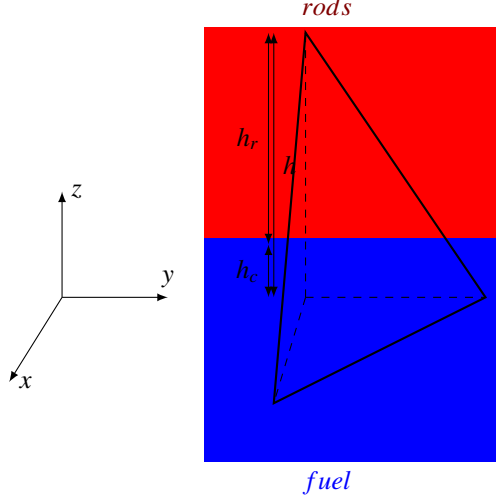


Figure 3: Heaviside function for the rod and the fuel on one tetrahedron.

197 *3.3. Acceleration using parareal in time Algorithm*

In order to accelerate the resolution of Equation (13) we apply the parareal in time algorithm [5] which is based on prediction correction scheme. We present in what follows the basic tools for applying such scheme. Let us divide the time interval $[0, T]$ into \hat{n} small ones;

$$[T_n, T_{n+1}] := \cup_{i=0}^{l-1} [T_{n,i}, T_{n,i+1}],$$

198 where $l = \frac{\Delta T}{\tau}$ with ΔT is the size of the small time interval we have $T_{n,0} = T_n$ et $T_{n,l} = T_{n+1}$.

199 Let $\{\mathbf{X}_n\}_{n \geq 0}$ be the sequence of solution of the system (13) at time $t = T_0, \dots, T_n, \dots, T_{\hat{n}}$,
 200 so that $\mathbf{X}_{n+1} := X_{n,l}$.

201 In the case of sequential propagation, those elements are solutions of:

$$\begin{bmatrix} \mathbf{Id} & & & & & \\ -\mathbf{F}_\Delta & \mathbf{Id} & & & & \\ & & \ddots & \ddots & & \\ & & & & -\mathbf{F}_\Delta & \mathbf{Id} \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_{\hat{n}-1} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad (16)$$

202 where $\mathbf{F}_\Delta = \prod_{i=1}^l E_i$ could be obtained with a “for”-loop as a routine and no need to
 203 memorize such results of matrix-product.

204 As well known The parareal in time algorithm [5, 6] ensures an exact convergence
 205 towards the fine approximated solution after \hat{n} iterations, where \hat{n} is the number of sub-
 206 intervals of the decomposition. And more accurate results with a time shorter when applied
 207 on diffusive systems. The updating formula for the series $(\mathbf{X}_n^p)_{n \geq 0}^{p \geq 0}$ of initial conditions val-
 208 ues is given as follows:

$$\mathbf{X}_{n+1}^{p+1} = \mathbf{G}_\Delta \mathbf{X}_n^{p+1} + \mathbf{F}_\Delta \mathbf{X}_n^p - \mathbf{G}_\Delta \mathbf{X}_n^p. \quad (17)$$

209 where \mathbf{G}_Δ is a coarse propagator taken with some reduction of the parallel propagator

210 \mathbf{F}_Δ that approximate the right solution with the θ -scheme using fine time step τ_F . This
 211 solvers reduction may produce some instability, one can find in [8, 10, 12] some stability
 212 results of such scheme. As the coarse part is solved sequentially, it is preferable to
 213 reduce model such that the coarse propagator \mathbf{G}_Δ is the least expensive as possible while
 214 remaining stable and always taking into account the physic problem. In what follows,
 215 we present a reduced neutron model that makes possible use of larger time step serial
 216 propagator.

217 4. Reduced neutron model and input experiments data

218 In practice, we consider some reduction of the model. This reduction will relate
 219 mainly to the dynamic aspect of rods in the nuclear reactor.

220 Let us describe the physics. At time $t = 0$ sec the first group of rods has an initial
 221 position at $z = 100$ cm, while the second group has a higher position at $z = 180$ cm.

222 The linear movement of rods is as follows:

223 Group 1 : ($t = 0$ | $z = 100$ cm) ↗ ($t = 26,5$ | $z = 180$ cm), with velocity of 3 cm/sec.

224 Group 2 : ($t = 7,5$ | $z = 180$ cm) ↘ ($t = 47,5$ | $z = 60$ cm), with velocity of 3 cm/sec.

225
 226 We present now both dynamics and statics scenarios which definitions are strongly
 227 related to the rods movement, and especially at their levels of sinking in the nuclear
 228 reactor. We show that a simplification of the model can be an advantage to accelerate
 229 the resolution by the parareal in time algorithm.

230 We call *complete-data model* and *reduced-data model* the neutron model i.e. Equa-
 231 tion (7) with and without precursors concentration groups respectively.

232 A thorough study of the reduction of the model is fundamental in order to allow a
 233 degradation of the fine solver and to determine the characteristics of a stable propagator.
 234 Degradation can be carried out either by some de-refinement of the mesh-grid (space-
 235 time), or by a reduction of the number of the iterations of inversion (when using an
 236 iterative solver) or by increasing the tolerance at the time of the resolution of the linear
 237 system or by simplification of the physics of the model. In what follows, we basically use
 238 this last way to build a coarse propagator.

239 In the framework of neutron kinetics, several parameters allow a simplification of the
 240 model while keeping the simulation results rather close to those obtained with a complete
 241 model. For this purpose, one can for example decrease the number of the groups of
 242 energies for the neutron flux, or even the number of the groups of the concentrations of
 243 the precursors. We can also approach the productivity of the reactor by a simple change
 244 of the reactivity factor.

245 Hereafter, we detail the dynamic aspect of the physical model and we propose a reduction
 246 of its dynamics. This reduction enables us to make computation cheaper while keeping
 247 accurate results as we show it in the experimental part later on (see section 5.3).

248 4.1. Dynamic rods scenario

249 The kinetics in this framework is characterized by a constant velocity (i.e. of 3cm/sec)
 250 of control rods. At the initial configuration, rods sinking level provides a critical state of
 251 the nuclear reactor (i.e. $k_{eff}=1$). In this configuration rods of group-1 are at $z = 100$ cm
 252 and group-2 are at $z = 180$ cm. During time the nuclear reactor is gradually switched

Algorithm 2: Algorithmme pararéel de la cinétique neutronique

```
Input:  $\hat{n} := \#slave\ proc, \tau_F, \tau_G$   
Input:  $\mathbf{X}_0^0 = [(\Phi^*)^T, (\mathbf{c}^*)^T]^T$  as initial conditions,  $\epsilon$  a tolerance of the algorithm ;  
Input: a solver  $\mathcal{A}$ , a data vector  $\mathbf{X}$ ;  
Routine( $\mathcal{A}, \mathbf{X}$ )  
1) Positioning the absorber rods with respect to the dynamic chronology;  
2) Constructing matrices related to equations (10)-(11)-(12);  
3) Serial propagation of  $\mathbf{X}$  using  $\mathcal{A}$  with respect to its rods' scenario (the result is denoted by  $\mathcal{A}\mathbf{X}$ );  
  
end Routine;  
 $p \leftarrow 0$ ;  
repeat  
  if master processor then  
    foreach  $n \in \{0, \dots, \hat{n} - 1\}$  do  
      1) Call: Routine( $\mathbf{G}_\Delta, \mathbf{X}_n^p$ ) (i.e. coarse-serial propagation);  
      2)  
        if  $p = 0$  then  
          | repeat return to 1 with  $\mathbf{X}_{n+1}^p$  until  $n = \hat{n} - 1$   
          else  
          | Construct  $(\mathbf{X}_n^p)_{n \geq 1}$  with respect to relationship (17);  
          end  
        3) Send  $(\mathbf{X}_n^p, processor(n))$  ;  
    end  
  else  
    forall slave processor( $n/n \in \{0, \dots, \hat{n} - 1\}$ ) do  
      | Recv  $(\mathbf{X}_n^p, master\ processor)$ ;  
      | Call: Routine( $\mathbf{F}_\Delta, \mathbf{X}_n^p$ ) (i.e. fine-parallel propagation);  
      | Send  $(\mathbf{F}_\Delta \mathbf{X}_n^p, processor(n))$  ;  
    end  
  end  
  if master processor then  
    foreach  $n \in \{0, \dots, \hat{n} - 1\}$  do  
      | Recv  $(\mathbf{F}_\Delta \mathbf{X}_n^p, processor(n))$  ;  
      | Evaluate  $\epsilon_n^p = \|\mathbf{F}_\Delta \mathbf{X}_n^p - \mathbf{X}_{n+1}^p\|_{\rho(\mathbb{R}^{84p})}$ ;  
    end  
  end  
   $p \leftarrow p + 1$ ;  
  Broadcast (master processor,  $\epsilon_n^p$ );  
until  $\max_n \epsilon_n^p \leq \epsilon$  ;
```

253 on to produce neutrons by fission. This production is less important if the rods sinking
254 level on the engine is important. By this way, one controls the fission productivity on a
255 nuclear reactor.

256 *4.2. Static rods scenario*

257 In this scenario rods are fixed. Then we approach the last scenario by choosing the
 258 rods positions correctly.

259 In the framework of the dynamic scenario, there are two principal phases of the core:
 260 subcritical and supercritical. A reproduction of these phases (related to the states of the
 261 core) is completely possible when we can arbitrarily control the core neutrons production.

262 In what follows we propose two approaches that ensure those states with a fixed
 263 position of control rods.

264 *Special rods positioning.* In order to approximate the latest scenario, we choose only two
 levels for the two rods groups.

	$t \in [0, 20]$	$t \in]20, 80]$
Group 1	$z=160$ cm	$z=180$ cm
Group 2	$z=180$ cm	$z=60$ cm

Table 1: Control of the state of the nuclear core by a change of the positions of the rods.

265

266 *External variation of reactivity.* This approach does not change the initial configuration
 267 of the rods. Nevertheless we are able to reproduce the phenomenon ensured by rods
 268 positioning. Indeed, on a physical point of view: these positions correspond to a certain
 269 reactivity, it is thus sufficient to reproduce this reactivity (corresponding to a given factor
 of k_{eff}) to reproduce the same scenario.

	$t \in [0, 20]$	$t \in]20, 80]$
reactivity	10008.e-5	9998.e-5

Table 2: Control of the state of the nuclear core by a change of its reactivity.

270

271 **Remark 1.** In the case of static scenario, it is easy and practice to implement the second
 272 method. Indeed, one does not require several updates of the coefficients, contrary to the
 273 first method. Moreover, the method 4.2 facilitates the development and accelerates the
 274 execution of the algorithm resulting without any change with respect to the results of
 275 the first approach 4.1.

276 **Remark 2.** With the static scenario it is difficult to reproduce the results of the real
 277 scenario. This is clear, because we already lose the continuity of the dynamics. In fact,
 278 it has a direct impact on energy resulting from this model. Nevertheless, with the static
 279 scenario of the second method 4.2, we can exploit the reactivity to reach the same peak
 280 as that of the real scenario. The static version 4.1 related to the rods position does not
 281 allow this possibility.

282 **5. Numerical results**

283 To validate the method and to test its performance in terms of numerical accuracy and
 284 computational effort, we consider the 3D variation of the LMW reference test case, which
 285 can be characterized as an operation transient with large spatial variation involving a
 286 LWR type reactor [35]. This test case is a three dimensional quarter core benchmark with
 287 two energy groups and six precursor groups. Figure 4 show the geometry of the model
 288 reactor. The withdrawal of the rods of group-1 initiates a transient, which is terminated
 289 by the insertion of the rods group-2. The average power increases and reaches maximum
 290 values between 20 and 21 seconds, then decreases slightly under the influence of the
 291 insertion of the rod group-2, and crosses its initial value after 37 seconds.

292 The physical data are presented at Table 3 and supplemented by the Tables 4. The
 293 cross sections are presented in Table 3. Each Table presents the section in a specific
 294 medium. The neutron data relating to the section of scattering are presented in Tables 4.

295 For the implementation and the numerical experiments of the preceding methods,
 296 we used data of various cross sections given from: [35]. The numerical simulations were
 297 carried out on a parallel SGI shared-memories-machine, which has 64 processor given
 298 rhythm with 2.0GHz, 256 Go of shared memory and a communication network Numalink
 299 (15 GB/s). We thus exploited this machine massively in parallel with the use of the MPI
 300 library and the scientific computation software FreeFem++[1].

301 This paragraph presents the various sections. The neutron flux domain is the sum
 302 of the two mediums of combustible (group-A and group-B), control rods and reflectors.
 Two energy groups are used for the discretization in each medium. The corresponding

Physical data	medium			
Cross sections	fuel A		fuel B	
	group-1	group-2	group-1	group-2
σ_t	0.23409670	0.93552546	0.23381787	0.95082160
σ_f	0.006477691	0.1127328	0.007503284	0.1378004
	rods		reflecteur	
	group-1	group-2	group-1	group-2
σ_t	0.23409670	0.93552546	0.20397003	1.26261670
σ_f	0.006477691	0.1127328	.0	.0
<i>celerity</i>	1.25e+7	2.5e+5	1.25e+7	2.5e+5.

Table 3: Neutron data for the core of benchmark 3D.

303 neutrons are in permanent interaction. These neutrons can change from an energy group
 304 to another by a deceleration (shocks) or by a simple change of direction (diffusion).
 305

306 *5.1. The sequential solver*

307 Experiment are carried-out with various neutron models and shows that the explicit
 308 Euler scheme is unconditionally unstable, while the θ -scheme related to a value of $0 <$
 309 $\theta < 1$ presents some oscillations with model when precursors concentration groups are
 310 present. It appears that the single value for which the θ -scheme is unconditionally stable
 311 and does not present oscillations is the Implicit Euler scheme i.e. $\theta = 1$. We present in

fuel A		
$\sigma_s^{(g \rightarrow g')}$	group-1	group-2
group-1	.20613914	.01755550
group-2	.0	.84786329
fuel B		
$\sigma_s^{(g \rightarrow g')}$	group-1	group-2
groupe-1	.20564756	.01717768
groupe-2	.85156526	.0
rods of control		
$\sigma_s^{(g \rightarrow g')}$	groupe-1	groupe-2
groupe-1	.20558914	.01755550
groupe-2	.84406329	.0
reflector		
$\sigma_s^{(g \rightarrow g')}$	groupe-1	groupe-2
groupe-1	.17371253	.02759693
groupe-2	1.21325319	.0

Table 4: "scattering" cross section data.

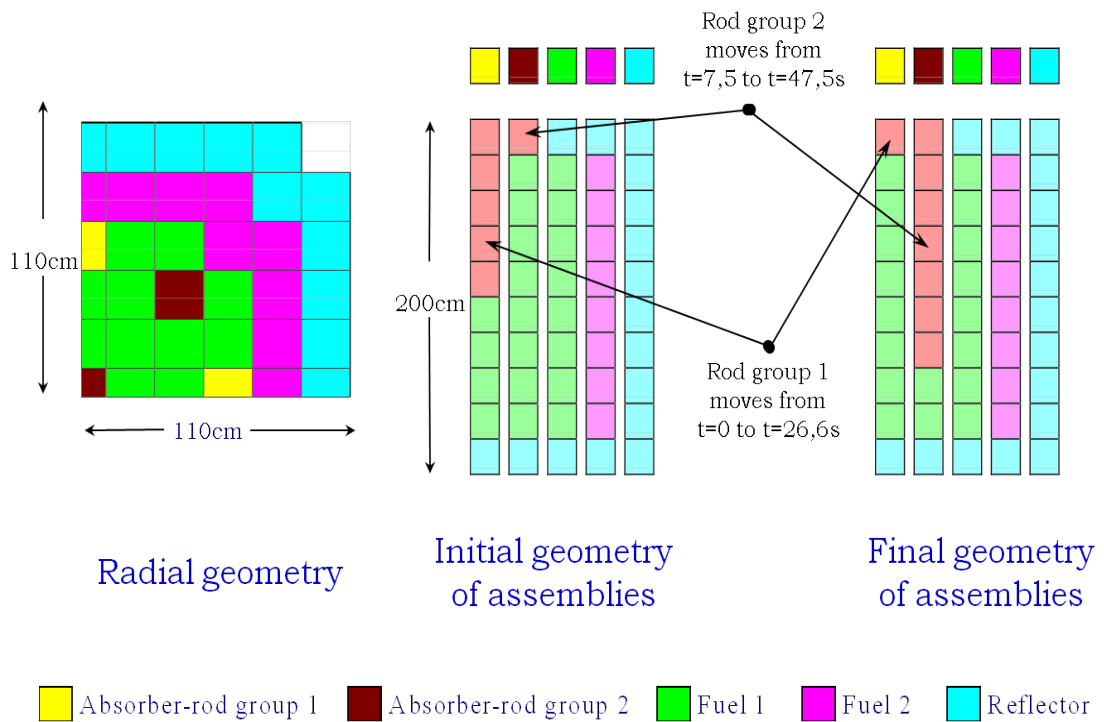


Figure 4: Initial configuration of rods as well as fuel volumes inside nuclear core .

312 what follows some graphs showing the behavior of the standard of the neutron flux in
 313 the nuclear reactor.

314 First of all, we simulate a complete neutron model, with six groups of precursors and
 315 two groups of neutron energy of flux. We eliminate then precursors group and consider
 316 the impact of their presence on the behavior of the reactor power.

317 *5.2. Parareal in time algorithm behavior with respect to the standard of the average flux*
 318 *on nuclear core*

319 In this subsection, we present a series of curves (see Figures 5 and 6) describing the
 state of the standard of the neutron flux in the engine in real time of the reaction.

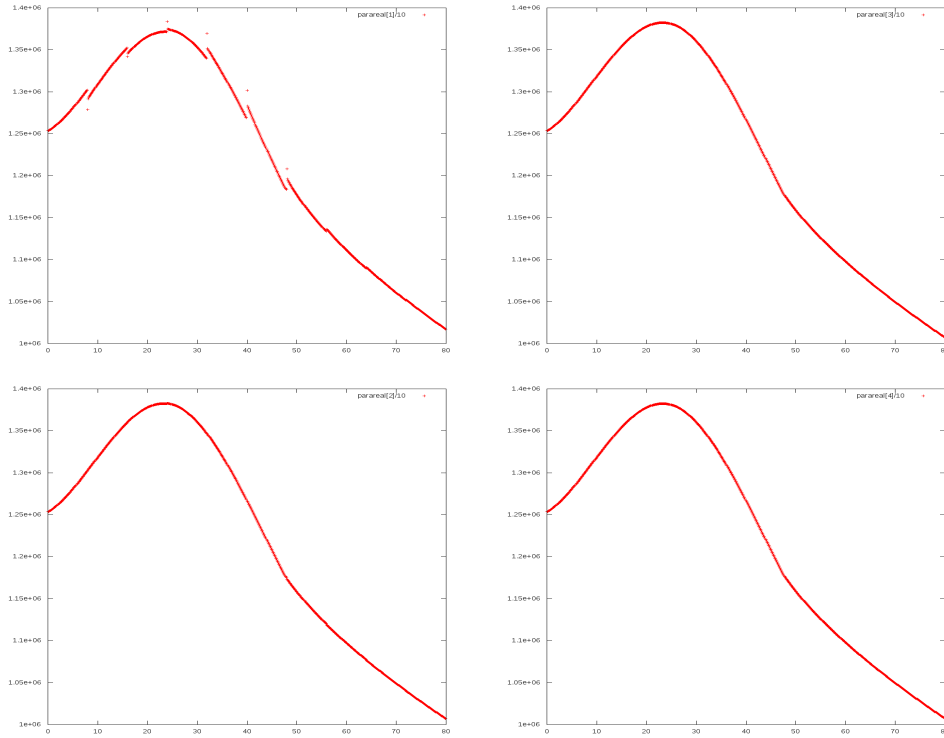


Figure 5: Parareal in time algorithm behavior applied on neutron model (1-4 iteration). Complet model with dynamic rods scenario $\tau_F = 10^{-1}$, $\tau_G = 4$ et $\hat{n} = 10$.

320

321 *5.3. Numerical study of the convergence of the parareal in time algorithm*

In this subsection we show the numerical convergence of the solutions calculated with the parareal in time algorithm to the sequential solution (see Tables 5-6-7-8) of which we vary the physics we treat. The error considered in these curves is the maximum one among ϵ_n^p , where :

$$\epsilon_n^p := \frac{\|\mathbf{X}_n^p - X_n\|_{L^2(\mathbb{R}^{8(q+p)})}}{\|X_n\|_{L^2(\mathbb{R}^{8(q+p)})}}.$$

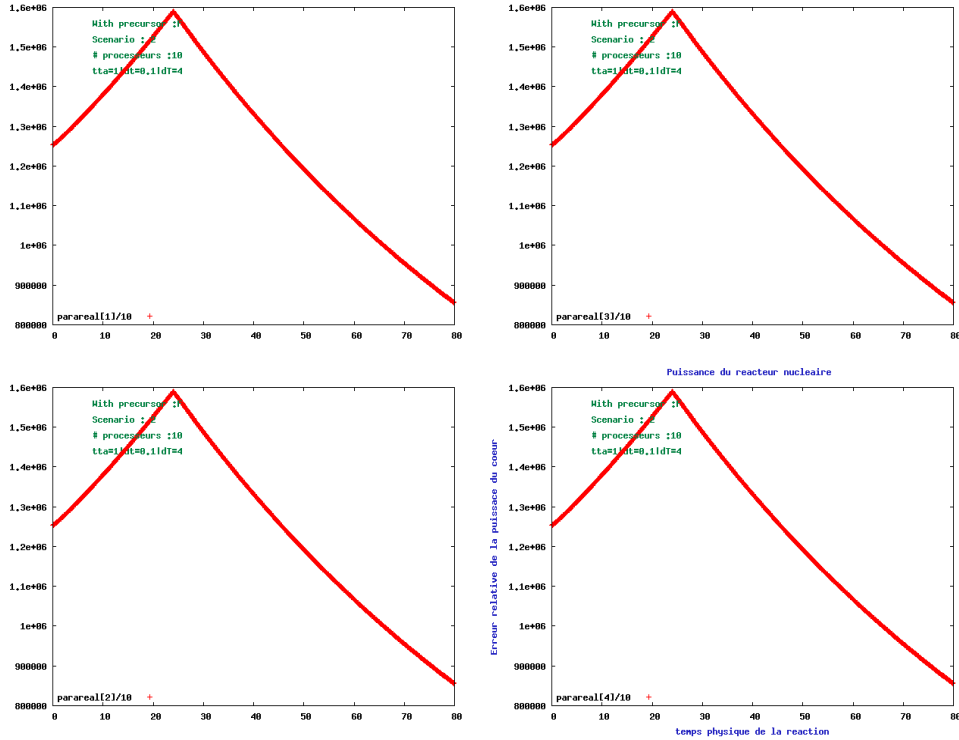


Figure 6: Parareal in time algorithm behavior applied on neutron model (1-4 iteration). Complet model with static rods scenario. $\tau_F = 10^0$, $\tau_G = 4$ et $\hat{n} = 10$.

322 We first present curves relating to the dynamic scenario with the complete model. In the
 323 second time, we present error curves obtained with the static scenario of a reduced-data
 324 model.

325 Taking into account the accuracy of the time-discretization scheme against referred to
 326 solution (reproduced with a very small time step e.g. 10^{-2}). We were thus interested in
 327 errors of the same order as this one. It is thus necessary to consider a convergence of the
 328 parareal in time algorithm to an error of order 10^{-2} or of 10^{-3} . Nevertheless, we have
 329 more thorough results of convergence (see Figure 7–10–11) to show convergence towards
 330 the sequential numerical solution.

331 **Remark 3.** We note that the error curves in Figure 7 (and in Figure 10 respectively)
 332 correspond to the experiment results of the behavior of the flux norm presented in Fig-
 333 ure 5 (respectively in Figure 6)

334 5.4. Reduced coarse solver using static rods scenario

335 We present in this sub-section a degradation of the coarse model allowing an ac-
 336 celeration of the resolution without harming the convergence of the algorithm. This
 337 degradation is carried out for the coarse solver at which we use static scenario. This

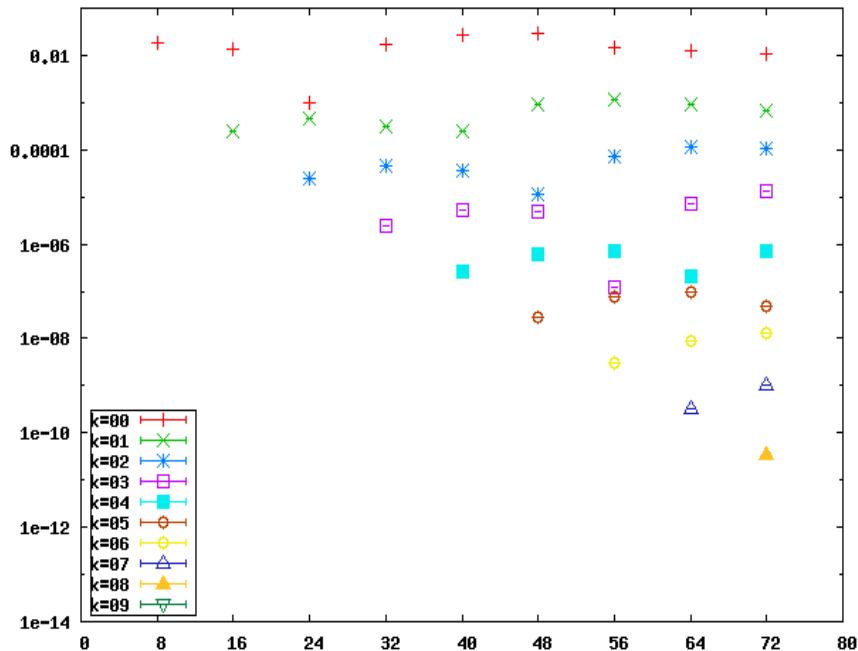


Figure 7: Convergence of the algorithm (iteration 00-09), complete model with dynamic rods scenario for both serial and parallel computation; $\tau_F = 10^{-1}$, $\tau_G = 4$ et $\hat{n} = 10$.

338 scenario as explained before uses a variation of the reactivity of the system on two inter-
 339 vals of physical time of the reaction. The employed procedure enables us to accelerate
 340 calculation by not computing the matrix of production at all time steps. Indeed, only one
 341 multiplication per a real coefficient of the matrix (already in memory) is enough.

342 To have an order of idea, the sequential simulation (execution with only one processor)
 343 lasts **03h:06mn:57s**.

344 We present in Figure 11 the acceleration of the resolution according to the number
 345 of subdivisions by temporal sub steps. The resolution on each sub steps is entrusted to a
 346 processor of the machine massively parallel. As we are interested in the errors of conver-
 347 gence about 10^{-2} or about 10^{-3} we note a saving of parallel time CPU of treatment of the
 348 neutron kinetics compared to a sequential calculation. Moreover, the fact of passing from
 349 8 processors to 16 processors makes it possible to pass from $2h : 45mn$ with $1h : 15mn$
 350 of treatment. We notice a division by two (approximate) of the time CPU, which is in
 351 adequacy with a local effectiveness of a machine with 8 proc the USSR by processor.

352 Notes that if the number of the subdivisions \hat{n} increases then the number of the pro-
 353 cessors agents in parallel also increases. The communication thus becomes increasingly
 354 important. This explains the fact that the curve of error in time wallclock (CPU) relating
 355 to the use of 40 processors is worse than that for 20 processors. The machine on which
 356 we submitted these numerical tests thus reached its saturation of speedup.

Table 5: Iteration 1 & 2 of the algorithm, complet model with dynamic rods scenario.

τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^1$	τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^2$
0.01	2	1.42e-02	0.01	2	3.04e-04
0.01	4	2.93e-02	0.01	4	1.16e-03
0.01	8	6.09e-02	0.01	8	4.11e-03
0.1	0.5	0.29e-02	0.1	0.5	1.31e-05
0.1	1	0.65e-02	0.1	1	6.48e-05
0.1	2	1.35e-02	0.1	2	0.28e-04
0.1	4	2.88e-02	0.1	4	1.11e-03
0.1	8	6.04e-02	0.1	8	4.02e-03
0.5	2	1.13e-02	0.5	2	1.71e-04
0.5	4	2.68e-02	0.5	4	0.88e-03
0.5	8	5.83e-02	0.5	8	3.55e-03
1	2	0.75e-02	1	2	7.42e-05
1	4	2.29e-02	1	4	0.63e-04
1	8	5.43e-02	1	8	3.01e-03

Table 6: Iteration 3 & 4 of the algorithm, complet model with dynamic rods scenario.

τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^3$	τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^4$
0.01	2	3.04e-04	0.01	2	1.73e-05
0.01	4	1.16e-03	0.01	4	1.22e-04
0.01	8	4.11e-03	0.01	8	7.85e-04
0.1	0.5	1.31e-05	0.1	0.5	1.41e-07
0.1	1	6.48e-05	0.1	1	1.67e-06
0.1	2	2.81e-04	0.1	2	1.51e-05
0.1	4	1.11e-03	0.1	4	1.13e-04
0.1	8	4.02e-03	0.1	8	7.53e-04
0.5	2	1.71e-04	0.5	2	7.02e-06
0.5	4	8.80e-04	0.5	4	7.80e-05
0.5	8	3.55e-03	0.5	8	6.17e-04
1	2	7.42e-05	1	2	1.94e-06
1	4	6.30e-04	1	4	4.64e-05
1	8	3.01e-03	1	8	4.77e-04

Table 7: Iteration 1 & 2 of the algorithm, reduced model with dynamic rods scenario.

τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^1$	τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^2$
0.1	2	1.39e-02	0.1	2	5.19e-05
0.1	4	2.78e-02	0.1	4	1.88e-04
0.1	8	5.34e-02	0.1	8	5.64e-04
0.5	2	1.09e-02	0.5	2	3.17e-05
0.5	4	2.48e-02	0.5	4	1.47e-04
0.5	8	5.03e-02	0.5	8	4.91e-04
1	2	7.22e-03	1	2	1.33e-05
1	4	2.10e-02	1	4	1.03e-04
1	8	4.65e-02	1	8	4.06e-04

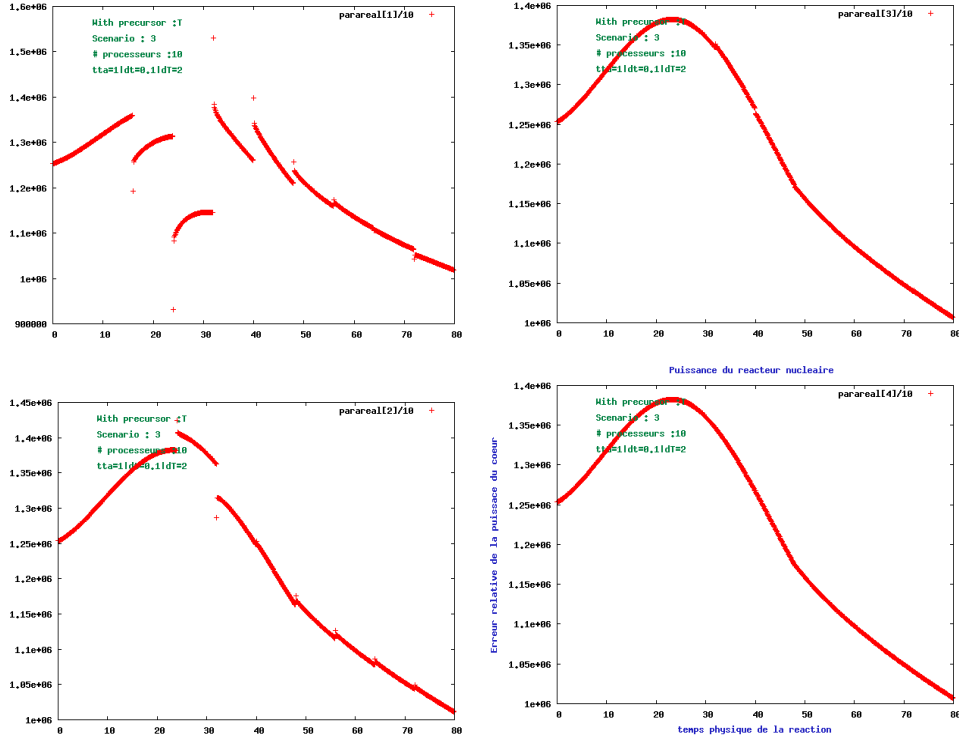


Figure 8: Norm of flux, behavior against (iteration 1-2-3-4) of the algorithm with complete model using static rods scenario for the coarse solver and dynamic rods scenario for the fine solver; $\tau_F = 10^{-1}$, $\tau_G = 2$ and $\hat{h} = 10$.

357 **Remark 4.** The parareal in time algorithm is implemented with a master-slave configuration
 358 for which we have two types of communications: a *distribution* and a *collection*
 359 communication. In the *distribution* communication, the main processor sends information
 360 towards all its processors agents. On the other hand in the *collection* communication;
 361 the Master himself receives and collects information since his agents. In both cases, it is
 362 about the same quantity of informations which passes in the two directions. The second
 363 type of communication is devoted to the correction of the coarse error, which requires
 364 fine information to be communicated by agent processors toward master. Those commu-
 365 nications are important and influence directly the period of information's passing. This
 366 is clear, in particular, if one increases the number of agent processors or the number of
 367 subintervals.

368 6. Conclusion

369 In this paper we present a time parallelization of the simulation of the neutron model,
 370 which is governed by the Boltzmann PDEs. The parallelization in time direction is

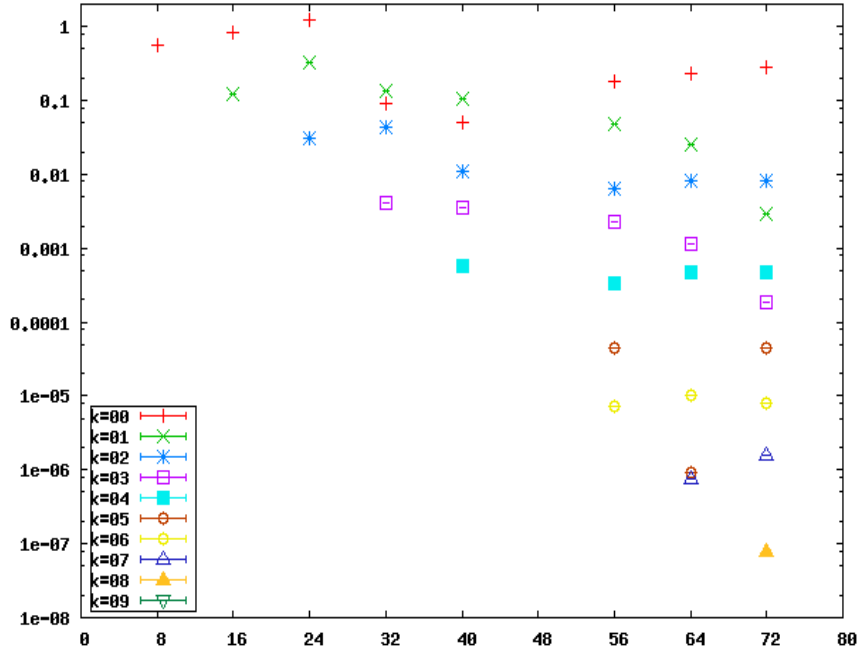


Figure 9: Convergence of the algorithm (iteration 00-09), complete model with dynamic rods scenario for the parallel computation and static rods scenario with reduced model for the coarse serial computation; $\tau_F = 10^{-1}$, $\tau_G = 2$ and $\hat{n} = 10$.

371 achieved with parareal in time algorithm. We showed that the neutron model could be
 372 reduced, in number of concentration of delayed neutrons also in the dynamic of the geom-
 373 etry, in such way the coarse propagator of the parareal in time algorithm is lightweight
 374 and can predict solution i.e. spatial flux distribution with a higher speedup. Furthermore,
 375 in the numerical point of view we can consider space domain decomposition (e.g. multi
 376 grid) in order to more reduce the model. This direction was investigated in the static
 377 case (see [2]) and is under study with more complicated neutron model. In the experi-
 378 ments point of view, we showed convergence rate and performance of the parallelization
 379 using supercomputer with MPI implementation of the parareal in time algorithm. The
 380 efficiency of the parallelization could be much more enhanced with some particular im-
 381 plementation using MPI non-blocking communications and managing agent processor's
 382 topology in a supercomputer.

Table 8: Iteration 3 & 4 of the algorithm, reduced model with dynamic rods scenario..

τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^3$	τ_F	τ_G	$\max_{n \geq 0} \epsilon_n^4$
0.1	2	1.91e-07	0.1	2	5.71e-09
0.1	4	1.74e-06	0.1	4	1.52e-07
0.1	8	1.63e-05	0.1	8	2.53e-06
0.5	2	1.01e-07	0.5	2	2.69e-09
0.5	4	1.30e-06	0.5	4	7.39e-08
0.5	8	1.21e-05	0.5	8	2.17e-06
1	2	3.18e-08	1	2	6.96e-10
1	4	8.35e-07	1	4	5.50e-08
1	8	9.42e-06	1	8	1.86e-06

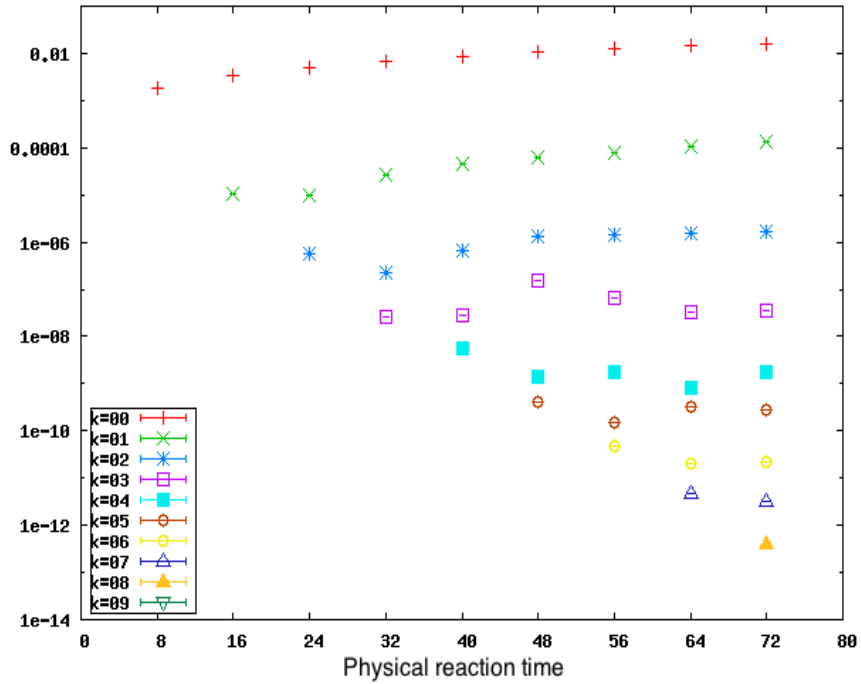


Figure 10: Convergence of the algorithm (iteration 00-09), complete model with dynamic rods scenario for the parallel computation and static rods scenario with reduced model for the coarse serial computation; $\tau_F = 10^{-1}$, $\tau_G = 4$ and $\hat{n} = 10$.

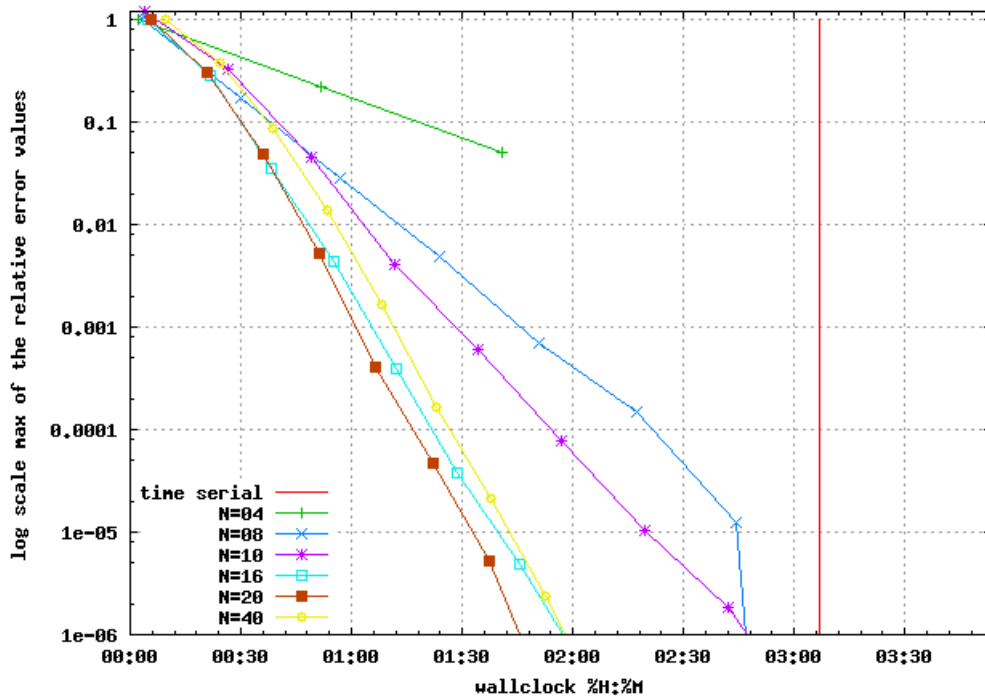


Figure 11: Real time machine simulation CPU (Wallclock format: h:m). Relative errors between solution computed with the parareal in time algorithm and the serial one, we vary the number of used processor per subinterval; $\tau_F = 10^{-1}$, $\tau_G = 2$ and $\hat{n} \in \{1, 4, 8, 10, 16, 20, 40\}$.

383 **References**

- 384 [1] O. Pironneau, F. Hecht, K. Ohtsuka, FreeFem++-mpi, third-edition:2012.
- 385 [2] P. Guérin, A.-M. Baudron, J.-J. Lautard, Domain decomposition methods for the neutron dif-
386 fusion problem, *Math. Comput. Simulation* 80 (11) (2010) 2159–2167, ISSN 0378-4754, doi:
387 10.1016/j.matcom.2010.04.009, URL <http://dx.doi.org/10.1016/j.matcom.2010.04.009>.
- 388 [3] M. Dahmani, A. Baudron, J. Lautard, L. Erradi, A 3D nodal mixed dual
389 method for nuclear reactor kinetics with improved quasistatic model and a semi-
390 implicit scheme to solve the precursor equations, *Annals of Nuclear Energy*
391 28 (8) (2001) 805–824, ISSN 0306-4549, doi:10.1016/S0306-4549(00)00089-X, URL
392 <http://www.sciencedirect.com/science/article/pii/S030645490000089X>.
- 393 [4] S. Chauvet, Multi-scale method for the resolution of the neutronic kinetics equations, Ph.D. thesis,
394 URL <http://tel.archives-ouvertes.fr/tel-00348435>, 2008.
- 395 [5] J.-L. Lions, Y. Maday, G. Turinici, Resolution d'EDP par un schema en temps "parareel", C.
396 R. Acad. Sci. Paris Ser. I Math. 332 (7) (2001) 661–668, ISSN 0764-4442, doi:10.1016/S0764-
397 4442(00)01793-6, URL [http://dx.doi.org/10.1016/S0764-4442\(00\)01793-6](http://dx.doi.org/10.1016/S0764-4442(00)01793-6).
- 398 [6] Y. Maday, G. Turinici, The parareal in time iterative solver: a further direction to paral-
399 lel implementation, in: *Domain decomposition methods in science and engineering*, vol. 40 of
400 *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 441–448, doi:10.1007/3-540-26825-1_45, URL
401 http://dx.doi.org/10.1007/3-540-26825-1_45, 2005.
- 402 [7] G. Bal, Y. Maday, A "parareal" time discretization for non-linear PDE's with application to the
403 pricing of an American put, in: *Recent developments in domain decomposition methods* (Zürich,
404 2001), vol. 23 of *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 189–202, 2002.
- 405 [8] G. Bal, On the convergence and the stability of the parareal algorithm to solve partial dif-
406 ferential equations, in: *Domain decomposition methods in science and engineering*, vol. 40 of
407 *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 425–432, doi:10.1007/3-540-26825-1_43, URL
408 http://dx.doi.org/10.1007/3-540-26825-1_43, 2005.
- 409 [9] M. J. Gander, E. Hairer, Nonlinear convergence analysis for the parareal algorithm, in: *Domain*
410 *decomposition methods in science and engineering XVII*, vol. 60 of *Lect. Notes Comput. Sci. Eng.*,
411 Springer, Berlin, 45–56, doi:10.1007/978-3-540-75199-1_4, URL http://dx.doi.org/10.1007/978-3-540-75199-1_4, 2008.
- 412 [10] D. S. Daoud, Stability of the parareal time discretization for parabolic inverse problems,
413 in: *Domain decomposition methods in science and engineering XVI*, vol. 55 of *Lect.*
414 *Notes Comput. Sci. Eng.*, Springer, Berlin, 275–282, doi:10.1007/978-3-540-34469-8_32, URL
415 http://dx.doi.org/10.1007/978-3-540-34469-8_32, 2007.
- 416 [11] M. J. Gander, S. Vandewalle, Analysis of the parareal time-parallel time-integration method, *SIAM*
417 *J. Sci. Comput.* 29 (2) (2007) 556–578 (electronic), ISSN 1064-8275, doi:10.1137/05064607X, URL
418 <http://dx.doi.org/10.1137/05064607X>.
- 419 [12] G. A. Staff, E. M. Rønquist, Stability of the parareal algorithm, in: *Domain decomposition methods*
420 *in science and engineering*, vol. 40 of *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 449–456, doi:
421 10.1007/3-540-26825-1_46, URL http://dx.doi.org/10.1007/3-540-26825-1_46, 2005.
- 422 [13] M. Sarkis, C. E. Schaerer, T. Mathew, Block diagonal parareal preconditioner for parabolic optimal
423 control problems, in: *Domain decomposition methods in science and engineering XVII*, vol. 60 of
424 *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 409–416, doi:10.1007/978-3-540-75199-1_52, URL
425 http://dx.doi.org/10.1007/978-3-540-75199-1_52, 2008.
- 426 [14] C. Farhat, J. Cortial, C. Dastillung, H. Bavestrello, Time-parallel implicit integrators
427 for the near-real-time prediction of linear structural dynamic responses, *Internat. J. Numer.*
428 *Methods Engrg.* 67 (5) (2006) 697–724, ISSN 0029-5981, doi:10.1002/nme.1653, URL
429 <http://dx.doi.org/10.1002/nme.1653>.
- 430 [15] G. Bal, Q. Wu, Symplectic parareal, in: *Domain decomposition methods in science and engineering*
431 *XVII*, vol. 60 of *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 401–408, doi:10.1007/978-3-540-
432 75199-1_51, URL http://dx.doi.org/10.1007/978-3-540-75199-1_51, 2008.
- 433 [16] S. Ulbrich, Generalized SQP methods with "parareal" time-domain decomposition for time-
434 dependent PDE-constrained optimization, in: *Real-time PDE-constrained optimization*, vol. 3 of
435 *Comput. Sci. Eng.*, SIAM, Philadelphia, PA, 145–168, 2007.
- 436 [17] M. J. Gander, S. Vandewalle, On the superlinear and linear convergence of the parareal al-
437 gorithm, in: *Domain decomposition methods in science and engineering XVI*, vol. 55 of *Lect.*
438 *Notes Comput. Sci. Eng.*, Springer, Berlin, 291–298, doi:10.1007/978-3-540-34469-8_34, URL
439 http://dx.doi.org/10.1007/978-3-540-34469-8_34, 2007.
- 440

- 441 [18] L. He, The reduced basis technique as a coarse solver for parareal in time simulations, *J. Comput.*
442 *Math.* 28 (5) (2010) 676–692, ISSN 0254-9409.
- 443 [19] M. Gander, M. Petcu, Analysis of a Krylov subspace enhanced parareal algorithm for linear prob-
444 lems, in: Paris-Sud Working Group on Modelling and Scientific Computing 2007–2008, vol. 25 of
445 *ESAIM Proc.*, EDP Sci., Les Ulis, 114–129, 2008.
- 446 [20] M. J. Gander, Analysis of the parareal algorithm applied to hyperbolic problems using character-
447 istics, *Bol. Soc. Esp. Mat. Apl. SēMA* (42) (2008) 21–35, ISSN 1575-9822.
- 448 [21] Y. Maday, Parareal in time algorithm for kinetic systems based on model reduction, in: High-
449 dimensional partial differential equations in science and engineering, vol. 41 of *CRM Proc. Lecture*
450 *Notes*, Amer. Math. Soc., Providence, RI, 183–194, 2007.
- 451 [22] D. Guibert, D. Tromeur-Dervout, Adaptive parareal for systems of ODEs, in: Domain decomposi-
452 tion methods in science and engineering XVI, vol. 55 of *Lect. Notes Comput. Sci. Eng.*, Springer,
453 Berlin, 587–594, doi:10.1007/978-3-540-34469-8_73, URL [http://dx.doi.org/10.1007/978-3-540-](http://dx.doi.org/10.1007/978-3-540-34469-8_73)
454 [34469-8_73](http://dx.doi.org/10.1007/978-3-540-34469-8_73), 2007.
- 455 [23] P. F. Fischer, F. Hecht, Y. Maday, A parareal in time semi-implicit approximation of the Navier-
456 Stokes equations, in: Domain decomposition methods in science and engineering, vol. 40 of
457 *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, 433–440, doi:10.1007/3-540-26825-1_44, URL
458 http://dx.doi.org/10.1007/3-540-26825-1_44, 2005.
- 459 [24] M. L. Minion, A hybrid parareal spectral deferred corrections method, *Commun. Appl. Math.*
460 *Comput. Sci.* 5 (2) (2010) 265–301.
- 461 [25] L. Baffico, S. Bernard, Y. Maday, G. Turinici, G. Zérah, Parallel-in-time molecular-
462 dynamics simulations, *Phys. Rev. E* 66 (2002) 057701, doi:10.1103/PhysRevE.66.057701, URL
463 <http://link.aps.org/doi/10.1103/PhysRevE.66.057701>.
- 464 [26] Y. Maday, G. Rim, Couplage de l’algorithme pararéel avec quelques méthodes de décomposition de
465 domaine, Ph.D. thesis, Paris, 2011.
- 466 [27] Y. Maday, G. Turinici, A parareal in time procedure for the control of partial differential equa-
467 tions, *C. R. Math. Acad. Sci. Paris* 335 (4) (2002) 387–392, ISSN 1631-073X, doi:10.1016/S1631-
468 073X(02)02467-6, URL [http://dx.doi.org/10.1016/S1631-073X\(02\)02467-6](http://dx.doi.org/10.1016/S1631-073X(02)02467-6).
- 469 [28] Y. Maday, J. Salomon, G. Turinici, Monotonic parareal control for quantum systems, *SIAM J.*
470 *Numer. Anal.* 45 (6) (2007) 2468–2482 (electronic), ISSN 0036-1429, doi:10.1137/050647086, URL
471 <http://dx.doi.org/10.1137/050647086>.
- 472 [29] T. P. Mathew, M. Sarkis, C. E. Schaerer, Analysis of block parareal preconditioners for parabolic
473 optimal control problems, *SIAM J. Sci. Comput.* 32 (3) (2010) 1180–1200, ISSN 1064-8275, doi:
474 10.1137/080717481, URL <http://dx.doi.org/10.1137/080717481>.
- 475 [30] R. Dautray, J.-L. Lions, Analyse mathématique et calcul numérique pour les sciences et les tech-
476 niques. Tome 1, Collection du Commissariat à l’Énergie Atomique: Série Scientifique. [Collection of
477 the Atomic Energy Commission: Science Series], Masson, Paris, ISBN 2-225-80406-9, with the col-
478 laboration of Michel Artola, Marc Authier, Philippe Bénilan, Michel Cessenat, Jean-Michel Combes,
479 André Gervat, Hélène Lanchon, Bertrand Mercier, Claude Wild and Claude Zuily, 1984.
- 480 [31] R. Dautray, J.-L. Lions, Analyse mathématique et calcul numérique pour les sciences et les tech-
481 niques. Vol. 9, INSTN: Collection Enseignement. [INSTN: Teaching Collection], Masson, Paris,
482 ISBN 2-225-81303-5, Évolution: numérique, transport. [Evolution: numerical methods, transport],
483 Reprint of the 1985 edition, 1988.
- 484 [32] F. Brezzi, M. Fortin, Mixed and hybrid finite element methods, vol. 15 of *Springer Series in*
485 *Computational Mathematics*, Springer-Verlag, New York, ISBN 0-387-97582-9, 1991.
- 486 [33] J. E. Roberts, J.-M. Thomas, Mixed and hybrid methods, in: Handbook of numerical analysis, Vol.
487 II, *Handb. Numer. Anal.*, II, North-Holland, Amsterdam, 523–639, 1991.
- 488 [34] Y. Saad, M. H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsym-
489 metric linear systems, *SIAM J. Sci. Statist. Comput.* 7 (3) (1986) 856–869, ISSN 0196-5204, doi:
490 10.1137/0907058, URL <http://dx.doi.org/10.1137/0907058>.
- 491 [35] W. W. Langenbuch S., Maurer W., Coarse-Mesh Flux-Expansion Method for the Analysis of Space-
492 Time Effects in Large Light Water Reactor Cores, *Nucl. Sci. Eng.* 63 (1977) 437–456.